Improved Adversarially Regularized Auto-Encoders (ARAE) for Text

Sebastien Baur Srivatsan Srinivasan Bernd Huber

Abstract

Autoencoders are extremely powerful in learning representations and features for continuous structures such as images but it has been extremely challenging to make them learn a compact smooth representation for discrete spaces such as text. Adversarially Regularized Autoencoders(ARAE) attempted to solve this problem by jointly training a discrete structure autoencoder and a smooth generator whose code distributions are matched using adversarial training, thus creating a smooth latent code space for text generation. While ARAE had its fair share of performance merits in text generation and bested other contemporary methods on many standard quantitative metrics, we identified issues such as unstable training, gradients vanishing and a notable performance drop with increasing sentence length. Addressing a few of these avenues, we were able to push the frontier of baseline significantly as measured by standard metrics, through a mix of better Wasserstein critic training methods, well-engineered constraints on the neural networks and the latent codes. Also, we pose several open questions from our suite of experiments which can drive future research in making ARAEs even more robust. The code used in this project can be found in https://github.com/harvard-ml-courses/cs287-s18-sb_ss

1. Introduction

Generative models play an important role in NLP because of their ability to effectively learn language structure from unlabeled data. Auto-regressive Language Models using recurrent architectures such as LSTM and GRU, have led the way in neural generative modeling, tasting tremendous success in tasks such as machine translation and speech recognition (Sutskever et al., 2014; Graves et al., 2013) . Yet, there are limitations to the probability structures that these models can learn and their inability to generate sentences conditionally offer us less control in text generation. Besides, test time performance in these models suffer from the well-known exposure bias (Ranzato et al., 2016).

Of late, there has been significant focus on using latent variable models for text generation, which in principle allays few such shortcomings. In this framework, we expect the models to encode salient features of the sentence such as context, style or grammar through the latent variables. With these learned codes, we could then generate diverse sentences by perturbing these codes or sentences with desired properties via conditioning on the code. Latent models have also been shown to soften exposure bias (Yu et al. (2016)).

Regularized autoencoders, both of variational and denoising varieties (Kingma & Welling (2013), Vincent et al. (2010)) have shown great progress in learning smooth representations of continuous high-dimensional spaces such as images. Yet, translating their success to discrete structures such as text continues to be a challenge. Work on VAEs for text (Bowman et al. (2016)) has been plagued by difficulties in optimization, with the decoder degenerating into language models. Similarly, Generative Adversarial Nets(GANs) have tried to circumvent the discrete structure difficulty through policy gradient methods (Yu et al. (2016)) or Gumbel-Softmax tricks (Kusner & Hernández-Lobato (2016)), but both have tasted little success in learning robust representations directly.

Adversarially Regularized Autoencoder (ARAE) (Zhao et al., 2017), proposed to tackle these issues by jointly training a discrete-space encoder and a smooth continuous space generator, and ensuring similarity in the two code distributions through GAN training. The encoder was thus regularized to learn a smooth contracted code space that best captured the complex local dependencies that existed in the discrete text. The model was able to map similar inputs to similar neighborhoods in the code space and its claims were demonstrated through multiple interpolation and style transfer experiments.

While ARAE is a critical step forward in unsupervised text generation, we believe that the model definitely has avenues for improvement to achieve better robustness and scale. For instance, it under-performs in generating coherent long sentences and is not successful in matching higher moments of code distributions. The objective of this work is to identify ARAE's shortcomings quantified by certain metrics, and suggest alternatives to improve them. Specifically, we show that (1) an improved training objective and

(2) well-engineered constraints on the neural networks and code space are pivotal in tackling these concerns.

Our best model demonstrates a solid improvement of **11 reverse PPL** points over the basline ARAE model (Zhao et al., 2017). It also outscores the baseline on the corpus-BLEU metrics which we use to measure overall sentence quality across different lengths. It also exhibits better generator performance by enhanced **higher-order moment matching** between the codes. With all these improvements,the model still retains the code space to be smooth and compact and provides similar reconstruction performance.

2. Background

2.1. Autoencoder

An auto-encoder is a model composed of two neural networks, an encoder and a decoder, that are trained to collectively learn an identity map. An encoder $\text{enc}_\phi : \mathbb{X} \to \mathbb{Z}$ learns to map the inputs $\mathbf{x}$ to latent code $\mathbf{z}$. The decoder dec_ψ learns the conditional distribution $p(\mathbf{x}|\mathbf{z})$, attempting to reconstruct the input back over $\mathbb{X}$. For any loss function $\mathcal{L}$, the training objective of an autoencoder could be specified as follows

$$\min_{\psi,\phi} \mathcal{L}_{\text{rec}} = \min_{\psi,\phi} \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_\mathbf{x}} \mathcal{L}(\text{dec}_\psi(\text{enc}_\phi(\mathbf{x})), \mathbf{x}) \quad (1)$$

where $\mathbb{P}_x$ is the distribution of the input x over $\mathbb{X}$.
For discrete structures such as text, we note that the inputs will be of the form $\mathbf{x} \in \mathcal{V}^n$ where $\mathcal{V}$ is the vocabulary.

2.2. Generative Adversarial Nets(GAN)

2.2.1. GAN - Introduction

Adversarial generative networks(GAN) were first introduced in 2014 by (Goodfellow et al., 2014). GANs can be thought of as a two player minimax game between a generator network which generates synthetic outputs $G(\mathbf{z})$ from a latent distribution p_z and a critic alias discriminator network which delineates the true(x) and synthetic outputs(G(z)) through confidence scores. If p_x, p_z are true data and latent variable distributions respectively, then the minimax objective of the game could be formalized as

$$V(G,D) = \mathbb{E}_{\mathbf{x}\sim p_x(\mathbf{x})} \log D(x) + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})} \log(1 - D(G(z)))$$

The training of a GAN alternates between training the discriminator and the generator networks. It can be shown that when the discriminator network is trained to optimality, taking a gradient step to update the weights of the generator is equivalent to taking a gradient step to minimize the Jensen-Shannon(JS) divergence between the synthetic and the true distribution (Arjovsky et al., 2017).

2.2.2. Wasserstein GAN(WGAN)

Demonstrating that minimizing JS divergence in standard GAN training suffers from mode collapse, exhibits poor mode matching and instability in training , (Arjovsky et al., 2017) introduce Wasserstein GAN (WGAN) that instead minimizes the Wasserstein distance between the two distributions. This approach is a weaker form of convergence, preventing mode collapse because of continuity, while correlating well with the quality of samples.

The WGAN architecture requires the critic network's output functional form to be Lipschitz continuous. (Arjovsky et al., 2017) achieves this by clipping the weights of the network to finite bounds and ARAE(Zhao et al. (2017)) adopts the same approach in its attempts to ensure Wassertein distance agreement between the codes. Gulrajani et al. (2017) demonstrate that weight clipping is subject to vanishing gradients and poor moment matching and thus followed multiple approaches(Gulrajani et al. (2017), Miyato et al. (2018)) to better enforce Lipschitz continuity which we will describe in Section 4

2.3. Progressive / Curriculum Learning

Humans and animals learn much better when the examples are not randomly presented but organized in a meaningful order which boasts of a gradual increase in complexity(Bengio et al., 2015). We explore this aspect in our curriculum training experiments in ARAE, where the model is sequentially exposed to a progressive vocabulary(easy to hard) based on different aspects such as vocabulary size and parts of speech. Usually in the earlier phases of harder generation tasks(long sentences with rich vocabulary), critic easily outperforms the generator since generator starts learning from an arbitrary latent space and such progressive training levels the playing field between the two. This approach also has a strong parallel with Progressive GAN model for images Karras et al. (2018a) which sequentially trains GANs based on the same images with increasing resolutions, thus progressively learning finer features.

3. Related Work

Our work largely builds on the work of the first ARAE model for text (Zhao et al., 2017) and will form the only baseline model for all our results. We note that it is hard to compare across approaches in the world of text GANs/ auto-encoders since there is discrepancy in standard baselines and metrics across prominent works. Within ARAE, we try to obtain better performance by attacking the problem in two directions - engineering better training objectives for enforcing Wasserstein distance, enforcing better constraints on the neural networks training and code space

and progressive training of GANs.

On generating better training objectives for WGAN, Gulrajani et al. (2017) impose a Lagrangian penalty on input gradients to the discriminator training objective and (Miyato et al., 2018) (spectral)normalizes all the discriminator layers' weights to contain the function to be 1-Lipschitz continuous. Our work experiments with both these approaches and quantifies their marginal contributions to ARAE training. For progressive training of GANs in text, our work is largely inspired by nuances from (Bengio et al., 2015) which formalizes curriculum learning as a method to sequentially expose models to more complicated examples and Progressive GAN ((Karras et al., 2018b)) which trains GANs with images of increasing resolutions to produce stellar results in generating sophisticated images.

It is also important to attend to other approaches that directly try to GAN generators over the discrete text space through policy gradient approaches as ARAE in future could benefit from the tricks employed in these works. We already alluded to the difficulties of training GANs directly in discrete space in Section 1. Building on top of SeqGANs((Yu et al., 2016)), RankGAN((Lin et al., 2017)) turns the discriminator objective to a ranking function in order to provide a richer feedback to the generator. (Guo et al., 2017) introduced LeakGAN with a leaky discriminator, whose features are encoded into a latent variable and passed to generator as feedback. Other works forced discriminator output onto continuous spaces (Rajeswar et al., 2017) or controlled certain semantic aspects of generated sentences by using VAE to learn discriminators (Hu et al., 2017). All these approaches adopted different baselines but largely claimed more robust training performance and generation of longer coherent sentences than their baselines.

4. Model

4.1. ARAE

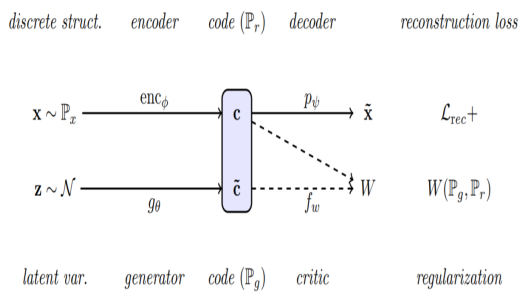


Figure 1. The ARAE architecture (Zhao et al. (2017))

Our model architecture is the same as the one used in Zhao et al. (2017), and is neatly described in Figure 1. We jointly train an AutoEncoder(AE) and a continuous smooth generator, both of whose codes are regularized using GAN training that tries to minimize Wasserstein Distance. The overall training objective(to be min.) could be summarized as :

$$\min_{\theta,\phi,\psi} \mathcal{L}_{rec}(\phi,\psi) + \lambda W(\mathbb{P}_r, \mathbb{P}_g)$$

where θ, ϕ, ψ are the parameters of the generator, the encoder, and the decoder respectively. $\mathcal{L}_{rec}$ is the cross entropy between the reconstructed and original input, at the token level (teacher forcing). W is the 1-Wasserstein distance between the codes from the discrete encoder model $\mathbb{P}_r$ and those from the generator $\mathbb{P}_g$. The model approximates W via an embedded critic function which is optimized adversarially to the encoder-generator pair. In practice, we alternate between minimizing these three losses:

- Reconstruction Loss of AE $\rightarrow \min_{\phi,\psi} \mathcal{L}_{rec}(\phi,\psi)$
- Training critic network f_ω to approximate the 1-Wasserstein distance (Theorem 5.10, Villani (2009))

$$\max_{\omega} \mathbb{E}_{x\sim\mathbb{P}_x} f_\omega(\text{enc}_\phi(x)) - \mathbb{E}_{c\sim\mathbb{P}_g} f_\omega(c) \quad (2)$$

- Training the generator and the encoder adversarially against the critic. Note that $\mathbb{P}_g$ is parametrized by θ.

$$\min_{\theta,\phi} -\mathbb{E}_{c\sim\mathbb{P}_g} f_\omega(c)$$

4.2. Lipschitz Continuity in ARAE

To approximate 1-Wasserstein distance with a critic, (Arjovsky et al., 2017) demonstrates that the function f_ω should be 1-Lipschitz continuous. Without proof, we state a lemma that helps us motivate the modifications to the critic training that we employed.

Lemma 1 *If $f(\theta)$ is differentiable everywhere and its gradients are bounded, we have the result :*

$$||f||_L \leq ||\nabla_\theta f||_\infty$$

where K -Lipschitz $||f||_L := \inf\{k > 0 : \forall x, y, ||f(x) - f(y)|| \leq K \cdot ||x - y||\}$

The Lemma could be summarized as -"Bounded gradients imply Lipschitz continuity". We next describe how different works use Lemma 1 to enforce Lipschitz continuity and also provide the corresponding experiment string tokens used in Section 6.

- **ARAE-WC(Weight Clipping) :** If the gradients are themselves continuous almost everywhere on a compact space, then they are also bounded. Therefore, clipping the weights ω of the critic network to a compact space enforces the Lipschitz continuity. This is the approach chosen by Arjovsky et al. (2017) and adopted in ARAE (Zhao et al., 2017).

- **ARAE-GP(Gradient Penalty) :** Adding a penalty on the input-gradients of the critic network to force them to stay close to one since we target 1-Lipschitz continuity. Gulrajani et al. (2017) proved that the solution of the Kantorovitch duality problem (Villani (2009)) has its gradients of norm 1 with probability 1 on all the segments linking points from the generated dataset to the true dataset. It suggests to add a constraint $(\mathbb{E}_{x\sim\mathbb{P}_{\tilde{x}}}||\nabla f_\omega(x)||_2 - 1)^2$ where $\mathbb{P}_{\tilde{x}}$ is the distribution of uniformly samples on the segments linking $\mathbb{P}_g$ to $\mathbb{P}_r$. Gulrajani et al. (2017) adopt this approach in improving WGANs.

- **ARAE-SN(Spectral Normalization) :** We can verify that $||f_\omega||_{Lip} \leq \prod_{l=1}^{n_L} \sigma(W_l)$, where σ is the spectral norm, since all neural network operations can be cast as composition of activation functions on matrix transformations of input vectors. Besides, common activation functions such as ReLU and Leaky-ReLU are themselves 1-Lipschitz continuous and Lipschitz Continuity is preserved on composition. This motivates the spectral normalization $\left(W \leftarrow \frac{W}{\sigma(W)}\right)$ of all critic layers' weights in Miyato et al. (2018). It is expected to have all the stability features of ARAE-GP but is computationally efficient due to fast approximation of spectral norm via power iteration method.

In line with observations of (Arjovsky et al., 2017), we foresaw two key shortcomings of ARAE-WC - a.) vanishing and exploding gradients, b.) poor moment matching at higher orders . We asserted that ARAE-GP helped address these claims through our demonstrations in Section 6

4.3. Constraint on the code

The ARAE architecture projects the generated code to unit-radius hypersphere. This was possibly done to prevent explosion or collapse of the code space, keeping it finite and compact. We also saw that ARAE-GP requires samples on segments linking $\mathbb{P}_g, \mathbb{P}_r$. Linear interpolation on the sphere can force us to calculate gradients on points outside the sphere where code is non-existent. Besides, enforcing the unit norm involves backpropagation through $x \rightarrow \frac{x}{||x||_2}$, whose gradients are unbounded and may explode to infinity. We observed both these issues when training ARAE-GP and propose the following modifications under a new experiment token **ARAE-GP-NP(Gradient Penalty and Code Norm Penalty)**.

- Using polar interpolation in the gradient penalty constraint, i.e. computing the expectation on points $\sqrt{u}x_r + \sqrt{1-u}x_g, u \sim \mathcal{U}(0,1), x_r \sim \mathbb{P}_r, x_g \sim \mathbb{P}_g$ instead of $ux_r + (1-u)x_g$. The first transformation, contrary to the second, almost preserves the norm.

- Replacing the unit norm constraint on the code with a penalty on the squared L_2 norm of the code in the encoder : $\lambda\mathbb{E}_{x\sim\mathbb{P}_x}||\text{enc}_\phi(x)||_2^2$

4.4. Curriculum learning

Karras et al. (2018a) introduced Progressive GAN as a way of systematically teaching GAN low-level to high level features by training the model on the same images across resolutions from 4x4 to 1024x1024. For text, an analogy to resolution could be the vocabulary. We therefore trained ARAE-GP progressively from a parsimonious to a richer vocabulary, using reconstruction accuracy thresholds to decide when to switch to the next training regime. We studied two such variants under the following experiment tokens:

- **ARAE-GP-PVOC(Progressive Vocabulary Size)** - Regimes based on vocabulary size(1000,2000,5000,8000,11000) with 'unk' tokens to replace out of vocabulary words.

- **ARAE-GP-PPOS(Progressive Part of Speech)** - Regimes based on Part of Speech(POS) Schedules with unique 'unk' tokens for each POS(Noun, Verb, Pron., Adj., Adv., Others). Appendix contains the exact POS schedule we used.

5. Methods

5.1. Dataset and Network Hyperparameters

We used a pre-processed version of the SNLI dataset containing 714k sentences written in English Bowman et al. (2015). Running all our proposed experiments on SNLI was already time and resource intensive and hence, we would leave experimenting with our best model on bigger datasets such as GigaWord (Napoles et al. (2012)) for future work. For the ARAE experiments, both the encoder and the decoder are one-layered LSTM with 300 and 500 hidden units respectively. The dimension of the code is 300. Both the generator and the critic are 3-layered MLP with default LeakyReLU activation and 500/300 hidden units per hidden layer, respectively. For the generator, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{100})$

5.2. Quantitative Metrics

Currently, there is still not a strong consensus on evaluation of sentence quality in unsupervised text generation using latent variable models. A key metric that we use is reverse perplexity(PPL) which is the perplexity we calculate by training a language model on the synthetic sentences and test them on true dataset. (Zhao et al., 2017) claims that reverse PPL is a good indicator of sentence quality and we observe the same. We also compute the true perplexity (SPPL -Synthetic PPL) using a synthetic language

represented by a language model (simple LSTM trained on SNLI that is used to generate a fake dataset with 700k sentences), similar to the evaluation done in Yu et al. (2016). Another metric that we use to show the performance across sentence length is to calculate corpus-BLEU scores for sentences bucketed by length and each hypothesis in our set of hypotheses(generated sentences) has the references of entire test set. This metric is again used to demonstrate performance in (Yu et al., 2016; Guo et al., 2017). Another key metric that we monitor is the reconstruction accuracy of the auto-encoder. We also report the moment-matching between the generator and auto-encoder code distributions to assert consistency between the codes.

5.3. Training and Inference

Our training and inference procedures follow the steps outlined in the original work.(Algorithm 1, Zhao et al. (2017)), training alternatively the GAN and the autoencoder. The autoencoder is trained using teacher forcing and SGD optimizer with constant learning rate of 1. For adversarial training, Arjovsky et al. (2017) suggests to eschew momentum based optimizers since the loss function is non-stationary, but we experimented with both ADAM (Kingma & Ba (2014)) and RMSProp (Dauphin et al. (2015)) and our experiments did not show any tangible differences in training performance and hence we stuck to the former. Another key trick to training WGANs is to train discriminator to optimality in order to provide meaningful signal for generator((Arjovsky et al., 2017)) and thus, we train the critic for more iterations than the generator at each step.

We use an early-stopping criterion based on reverse PPL, computed using a language model of order 5 (implemented with KenLM (Heafield (2011))). Most of our models converged after 3 epochs. When using norm penalty (ARAE-GP-NP) on the code instead of fixing the norm to 1, this usually takes much longer (7-17 epochs).

5.4. Qualitative Evaluation

To further illustrate the effectiveness of our approach and to ascertain that the model uses the code instead of degenerating to a language model, we generate sentences under the following experiment settings for human evaluation.

1. **Generating Sentences from Noise -** To provide a general sense of sentence quality, we sample each model for randomly generated sentences.

2. **Interpolation -** To further illustrate the smoothness of the code, we interpolate in the hidden space linearly between two randomly sampled codes. Specifically, we investigate smoothness by sampling two points z_0 and z_1 from $p(z)$ and constructing intermediary points $z(\lambda) = \lambda z_1 + (1 - \lambda)z_0$, linearly interpolating in the latent space. For each code, we generate the decoded sentence. For fair model comparison, we also provide aligned samples where we sample two fixed sentences, encode them to z_0, z_1 and generate sentences for the interpolations using the top candidate models.

3. **Noise Perturbation -**An ideal representation should be relatively robust to small changes through noise in the input (latent space). To test this, we generate samples with additive noise on the code, and verify that the output didn't alter radically under perturbations.

Dwelling on the code used for this project, we thank the authors of (Zhao et al., 2017), whose code repository (`https://github.com/jakezhaojb/ARAE`) on ARAE has formed the backbone for our development efforts and we built a substantial amount of new code on top of it for making all the proposed modifications, running experiments and evaluations.

6. Results

Model	RPPL	RCA(%)	SPPL
ARAE-WC	74.5	83	19.6
ARAE-GP	72.1	82	17.1
ARAE-SN	75.1	80	18.8
ARAE-GP-NP	**63.3**	75	**15.1**
ARAE-PVOC	77.0	79	NA
ARAE-PPOS	73.9	80	NA

Table 1. Table documenting key quantitative metrics in the six major experiments. **ARAE-GP-NP** seems to be the best performer considering trade-offs. RPPL - Reverse Perplexity, RCA - Reconstruction Accuracy, SPPL - Synthetic Perplexity (Refer Section 5.2 for definitions). Note that we did not compute synthetic PPL for progressive vocabulary experiments because of paucity of time.

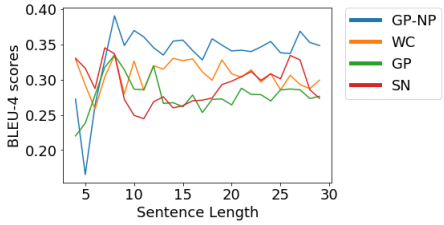


Figure 2. **BLEU scores**(y-axis) as a function of sentence length(x-axis) across 4 experiments (WC, GP, GP-NP, SN). BLEU is computed by using generated sentences(capped at 1000) of a given length as hypotheses, and the entire test set as references. We notice a performance dip for all models with longer sentences except **ARAE-GP-NP**. We believe that it is an effect of the better optima reached by both its generator (see Figure 3) and its decoder (0.76 vs 1.1 test loss).

In this section, we compare the models based on reverse PPL(the lower the better) as key driver metric. Table 1 shows all the metrics we used to evaluate the models. Figure 2 shows the corpus-BLEU scores of the model as function of sentence length, which we use to study the quality of the model as it generates longer sentences. Figure 3 shows moment matching between the code distributions of the generator and the encoder. On all these forms of evaluation, the model **ARAE-GP-NP** seems to emerge as the best performing model. We will explore finer facets of these results in this section through some discussion notes.

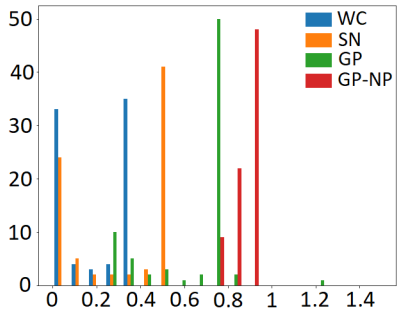
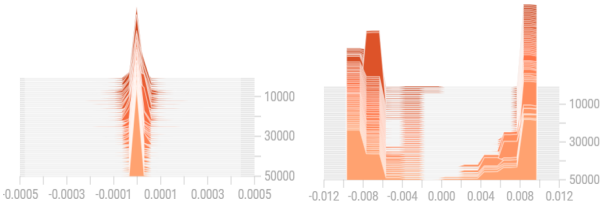


Figure 3. Relative moment matching $\frac{\mathbb{E}_{c\sim\mathbb{P}_g}|c|^k}{\mathbb{E}_{c\sim\mathbb{P}_r}|c|^k}, \forall k \in [1, 80]$. Ideally it should be all 1. Only GP-NP is able to match the moments in a satisfactory manner.

6.1. Experiments on better WGAN training

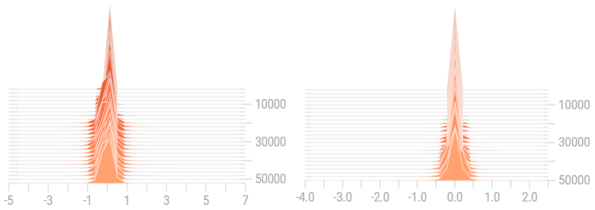
6.1.1. Issues with Weight Clipping

The weight clipping method is difficult to parametrize because it involves choosing the the bound on weights. We have $\omega \in [-c, c]^d \implies \exists K > 0 : ||f_\omega||_L = K$. But the constant K is unknown. In practice, this means that we are optimizing over $K \cdot W(\mathbb{P}_g, \mathbb{P}_r)$ instead of $W(\mathbb{P}_g, \mathbb{P}_r)$. It does not violate any fundamental Wasserstein assumptions but robs critic outputs of any possible interpretability and hence makes it hard to compare critic performance with different models. One other problem is that the way c is chosen has an influence on the dynamics of the network. The weights of different layers have a priori no reason to be of same scale. We observe that this choice leads to gradient vanishing: the critic output provides gradients of very low magnitude and hence each update of the weights of the generator is almost insignificant. Besides, the weights of the critic tend to accumulate around the bounds of the interval, which does not look very natural: the gradients push them outside $[-c, c]$ but they can not go outside this range and in the end the critic weights do not update. We can see both these issues in Figure 4.



((a)) ((b))

Figure 4. **ARAE-WC Training Issues :** a.) Vanishing gradients in the generator during training. b.)Weight of the last layer of the critic network during training. See that a lot of weights being cut off at the barriers.

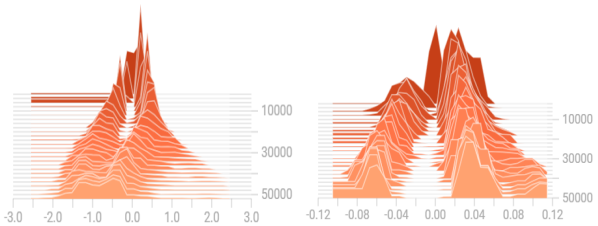


((a)) ARAE-GP ((b)) ARAE-SN

Figure 5. Gradient vanishing doesn't occur with GP, SN

6.1.2. Enhanced Critic Objectives : GP, SN

Spectral normalization (ARAE-SN) and gradient penalty (ARAE-GP) were earlier described to be better alternatives to enforce Lipschitz continuity. While ARAE-GP was definitely better than ARAE-WC, surprisingly ARAE-SN was not performing better(Check Table 1). Having said that, both methods get rid of the vanishing gradient problem (see Figure 5), but the weights of the last layer of the discriminator still look abnormal, especially for ARAE-SN (see Figure 6). We also want to note that ARAE-SN, ARAE-GP, and ARAE-WC have very similar training behavior in terms of reaching optimal reverse PPL within 2-3 epochs.

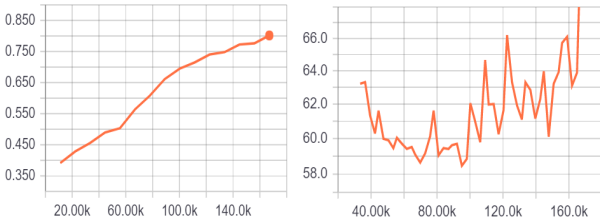


((a)) ARAE-GP ((b)) ARAE-SN

Figure 6. Weights of last layer of the critic network

6.2. Constraint on the code (ARAE-GP-NP)

ARAE-GP happened to yield the best results in terms of reverse PPL. For this reason we further experimented with this model, aiming at lowering the reverse PPL again and it gave birth to ARAE-GP-NP experiment. Not constraining the code on the unit sphere(done for making code compact and non-collapsible) but rather allowing the model to penalize(Lagrangian) the code's norm seems to be a more natural way of enforcing compactness. While ARAE-GP tends to reach its best performance after 3 epochs of training, and quickly deteriorate afterwards, ARAE-GP-NP took about 17 epochs(which we were able to reduce to 7 - Refer Section 7.4) to train well. Contrary to ARAE-GP, its reverse perplexity does not strictly increase after reaching its minimum, and stays around this value. Besides, the decoder also takes a lot of time to reach good reconstruction accuracy. Also, we saw that doing polar interpolation on the sphere than linear did not improve performance by much. The essence of these results is all captured in Figure 7. We believe that ideas like choosing better early stopping criteria and hyperparameter tuning for optimal Lagrangian Penalty multiple(λ) would help improve training of ARAE-GP-NP to produce comparable reconstruction accuracy to ARAE-GP while still retaining its impressive reverse PPL results.



((a)) Reconstruction accuracy ((b)) Reverse PPL

Figure 7. Performance of ARAE-GP-NP

6.3. Qualitative Sentence Sampling Experiments

In Appendix B, we can find the different experiments we described in Section 5.4, which are done largely to ascertain the smoothness and robustness to perturbations of the learned code space. We see that the models show some promise on random sampling and noise perturbation experiments, providing intuitive outputs. On the other hand, the sentence interpolations neither in the baseline nor the other models looked good semantically and looked just below par grammatically too. Also, the difference in reverse PPL between the models does not seem to translate into the same amount of quality difference between the sentences. (Cífka et al., 2018) suggest that while quantitative reconstruction metrics of ARAE are good, sampling quality was poor under human evaluation and it resonates with our observation that it was harder to distinguish the quality of sampled sentences across models as the sentence structure and semantics was uniformly unsatisfactory across the three.

6.4. Progressive vocabulary

We performed two experiments with progressive vocabulary - ARAE-PVOC (increasing vocabulary size) and ARAE-PPOS(increasing size by adding more from each POS tag). We switched between sizes when our reconstruction accuracy reaches more than 75%. As is intuitive, the more parsimonious vocabularies reached these cutoffs within the first epoch and the larger vocabularies took longer. Yet, our results with both these methods could not beat the performance of ARAE-GP baselines, which is surprising. Having said that ARAE-PPOS performs better than ARAE-PVOC since growing vocabulary is better captured by growing POS knowledge than just a vocabulary based on frequency of occurrence. We believe that the under-performance issues may lie with the choice of bins and cutoffs and or better/smoother vocabulary blending methods(say adding a new word every few forward passes than adding discretely a set of words after certain passes). An interesting future line of work particularly for the POS progression would be to employ a StackGAN architecture (Zhang et al. (2017)) to produce sentence in two stacked steps - first one producing a coarse sentence with POS-OOV tokens and the second layer using the output of the first to fill in the dubious POS-OOV tokens to finally generate coherent longer sentences.

7. Discussion

Section 6 described our key results. In this process, we had several interesting observations, open questions and directions for future work while running our experiments, which we bring to readers' attention with short discussion notes.

7.1. Training Tricks

7.1.1. Batch normalization

Batch normalization (Ioffe & Szegedy (2015)) standardizes the outputs of activation layers (essentially a covariance shift), increasing training stability by allowing easier backpropagation of gradients and reducing the sensitivity of performance on learning rate and initial weights. In our experiments, this translates to better training (both in terms of speed of convergence and final perplexity) and better stability (gradient explosion/vanishing are less likely). We experimented with batch normalization either in the generator only, in both the generator and the critic, or in none. It turns out that the model performed better with batch normalization on both components by $\sim$ **1 reverse PPL** point.

7.1.2. PENALTY ON DRIFT

We noticed that the output of the critic tends to drift to infinity through training, which is not a desirable behavior. To prevent it from happening, we added a term $10^{-3}\mathbb{E}_{c\sim\mathbb{P}_r} f_\omega(c)^2$ in the loss of the critic, an idea borrowed from Karras et al. (2018a). While this did not affect final performance tangibly, this stabilized training better by ensuring that the critic outputs remain bounded.

7.1.3. WORD DROPOUT

When the decoder is too powerful, there is a risk of overfitting largely to our training data which is undesirable for generalizability. To prevent from this from happening, we introduced a 50% word dropout in the decoder, so that it does not just learn the dataset. This resulted in a marginal decrease of 0.5 reverse PPL points.

7.1.4. CAPACITY OF THE MODEL

We tried replacing basic LSTM with bidirectional LSTM as an encoder and it again added a small marginal contribution to the model's performance. It is suggested that improving the capacity of the generator provides enough architectural power to learn complex multi-modal distributions which are common in text. Increasing the hidden dimension of the decoder and the generator from 300 to 500 led to small improvements in reverse PPL ($\sim$ **0.5 - 1** points).

7.2. ARAE-SN vs ARAE-GP

The under-performance of ARAE-SN as compared to ARAE-GP still remains an open question. If we use 1-Lipschitz activation functions such as ReLU, we already saw that Spectral Normalization should definitely make the critic network 1-Lipschitz continuous since the product of spectral norms of layer weights is an upper bound to the Lipschitz norm. We initially thought that the power iteration method (Miyato et al., 2018) did not approximate the norm of the weights well enough, but those assumptions were quelled as we ran spot tests of difference between true spectral norm and the fast approximation during random forward iterations and the differences were negligible (less than 0.001 relative error). Also, we check that the spectral norm of the normalized weights were tangibly indifferent from 1. Also, a strange observation we noticed in Figure 6 is that the distribution of weights in the last layer of the critic were odd for ARAE-SN as compared to ARAE-GP and we leave this interesting investigation of under-performance of ARAE-SN for future work.

7.3. ARAE-GP-NP Training

From our training we observed that ARAE-GP-NP during the initial few epochs learns to degenerate to be just a language model (which is an oversimplification since the reconstruction accuracy, while considerable, was lower than what ARAE-GP achieves after the same training time), and ends up using the code after it can no longer make any progress through just the language model. We also verified that the model used the code in the later epochs by asserting that both the code magnitude and the weights were non-zero. through our qualitative smoothness sampling experiments as the interpolations, while not the best,was just smooth enough when viewed under the same lens as the baselines. We present a weak argument as to why this model works much better than ARAE-GP on BLEU and reverse PPL, and this definitely warrants further deep-dive in future. Remember that ARAE-GP projects all the codes to the surface of the unit hyper-sphere. This means that for gradient penalty, we take interpolates on a spherical surface in ARAE-GP as compared to a L_2 ball in ARAE-GP-NP. In the former, the Lipschitz continuity of the critic is therefore enforced on a part of the space which does not correspond to any data point. We observe that the gradients tend not to have norm 1 on all over the sphere, which means that the ARAE-GP critic is a poorer approximation of the solution of the Kantorovitch Duality problem, and therefore the gradients passed to the generator by ARAE-GP-NP are a better approximation of the true gradients, which helps both the generator and the encoder to possibly converge to a better distribution mapping in the code space.

7.4. Artificial Noise with Encoder

In the original ARAE model, an additive centered Gaussian noise (with standard deviation starting at 0.2 and exponentially decreasing to 0 over time) is added to the output of the encoder. We believe that this may have been done to make the decoder more robust to the generated code. To mirror the findings of Cífka et al. (2018), we tried to implement a simple Adversarial Autoencoder (Makhzani et al. (2015)) which had alarmingly bad quantitative metrics. On digging deeper, we realized that the magnitude of this noise was pretty high and pushed the overall code's norm to above 2. The critic thus had an easy job of discriminating encoder's and generator's code (approx. unit norm) and it took a lot of epochs for the encoder's code to reach similar norms with the exponentially decaying noise. We then removed this noise, this resulted in dramatic performance gains for AAE in terms of reverse PPL. However, the performance was still $\sim$ 20 reverse PPL points worse than ARAE. We are tempted to beleive that this could be an effect of some implementation oversight. But the findings about the noise helped us improve the operational performance of ARAE-GP-NP which actually had a similar problem where the code learned was pretty small in overall magnitude ($\sim$ 0.1) and the noise seemed to be dominating the overall norm. This helped explain why ARAE-

GP-NP took about 17 epochs to achieve satisfactory performance on reconstruction accuracy. We validated that claim by running ARAE-GP-NP by removing the artificial noise and the experiment converged to similar PPL and reconstruction numbers within 7 epochs i.e. trained faster with the same performance efficiency. It will be an interesting extension to test this newly trained model (ARAE-GP-NP without encoder noise) on other quantitative and qualitative experiments that we performed for our top models.

7.5. Utilization of the code

Qualitative experiments with ARAE-GP-NP (Appendix B) show that conditional generation (i.e. encoding of a sentence, and decoding using only the code and its own outputs as inputs) works reasonably but is still far from ideal. The first few words usually make both syntactic and semantic sense, but as the sentence gets longer, the context and meaning deteriorate. We suspected that the model does not use the code strongly enough once it generates enough text and falls back almost exclusively to its language modeling mode. We explored this hypothesis by observing the gradients of the decoder with regard to its input. We observed that the gradients are very high for hidden state for the first few words (~ 5) at the start of the sequence, but as we keep moving further down the sequence, most information seems to be propagated only through the auto-regressive generation and the gradients for the latent code seem to be pretty low.

While it is logical that the code provides information for the decoder to generate the starting words of the sentence, we also hoped that the code would contain more global information such as the theme/context of the sentence for the decoder to leverage for long text generation with desirable constraints. For example, two sentences starting with *A woman wearing a black coat..* can be completed in many different ways based on other latent characteristics of the sentence. This reminds us that all our ARAE models still have several leaps to make for truly encoding entire long sentences on to a code space that implicitly understands different style and context components within those sentences in order to achieve superlative text generation performance.

8. Conclusion

In this work, we presented a suite of methods for making the ARAE model for generating text more robust. Through this work, we have highlighted certain open issues in ARAE and addressed a portion of them through novel constraints on the objective/loss functions, well-engineered training methods and network architectures. Recognizing that unsupervised text generation using latent variables has miles to go before producing the startling results that latent variable models do with images, one needs to look further than just the successful quantitative metrics that this work produces. This work also made us appreciate the sensitivity of ARAE training to training hyperparameters, model architectures, the impact of constraining the model components(such as norm of code) on overall training objectives and several other practical issues that make unsupervised text generation using ARAE a hard and challenging problem. Both the successful and failed experiments ask pertinent open questions and suggests interesting future directions to propel ARAE further towards more robust natural language generation.

Acknowledgements

We sincerely thank Prof. Alexander Rush, SEAS, Harvard University and Yoon Kim, SEAS, Harvard University for mentoring us throughout this project with apt guidance, valuable ideas and experiment designs and also offering us with adequate amount of computational resources to run experiments at scale.

References

Arjovsky, Martín, Chintala, Soumith, and Bottou, Léon. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pp. 214–223, 2017.

Bengio, Yoshua, Louradour, Jrme, Collobert, Ronan, and Weston, Jason. Curriculum learning, 2015.

Bowman, Samuel R, Angeli, Gabor, Potts, Christopher, and Manning, Christopher D. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

Bowman, Samuel R., Vilnis, Luke, Vinyals, Oriol, Dai, Andrew M., Józefowicz, Rafal, and Bengio, Samy. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pp. 10–21, 2016.

Cífka, Ondrej, Severyn, Aliaksei, Alfonseca, Enrique, and Filippova, Katja. Eval all, trust a few, do wrong to none: Comparing sentence generation models. *CoRR*, abs/1804.07972, 2018.

Dauphin, Yann N., de Vries, Harm, Chung, Junyoung, and Bengio, Yoshua. Rmsprop and equilibrated adaptive learning rates for non-convex optimization. *CoRR*, abs/1502.04390, 2015.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville,

Aaron, and Bengio, Yoshua. Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. Curran Associates, Inc., 2014.

Graves, Alex, Mohamed, Abdel-rahman, and Hinton, Geoffrey. Speech recognition with deep recurrent neural networks. *arXiv:1303.5778*, 2013.

Gulrajani, Ishaan, Ahmed, Faruk, Arjovsky, Martin, Dumoulin, Vincent, and Courville, Aaron C. Improved training of wasserstein gans. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 5767–5777. Curran Associates, Inc., 2017.

Guo, Jiaxian, Lu, Sidi, Cai, Han, Zhang, Weinan, Yu, Yong, and Wang, Jun. Long text generation via adversarial training with leaked information. *arXiv preprint arXiv:1709.08624*, 2017.

Heafield, Kenneth. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, pp. 187–197, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-12-1.

Hu, Zhiting, Yang, Zichao, Liang, Xiaodan, Salakhutdinov, Ruslan, and Xing, Eric P. Toward controlled generation of text. In *International Conference on Machine Learning*, pp. 1587–1596, 2017.

Ioffe, Sergey and Szegedy, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 448–456. JMLR.org, 2015.

Karras, Tero, Aila, Timo, Laine, Samuli, and Lehtinen, Jaakko. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018a.

Karras, Tero, Aila, Timo, Laine, Samuli, and Lehtinen, Jaakko. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2018b.

Kingma, Diederik P. and Ba, Jimmy. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

Kingma, Diederik P. and Welling, Max. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013. URL http://dblp.uni-trier.de/db/journals/corr/corr1312.html#KingmaW13.

Kusner, Matt J. and Hernández-Lobato, José Miguel. GANS for sequences of discrete elements with the gumbel-softmax distribution. *CoRR*, abs/1611.04051, 2016.

Lin, Kevin, Li, Diangi, He, Xiadong, Zhang, Zhengyou, and Sun, Ming-Ting. Adversarial ranking for language generation. *arXiv:1705.11001*, 2017.

Makhzani, Alireza, Shlens, Jonathon, Jaitly, Navdeep, and Goodfellow, Ian J. Adversarial autoencoders. *CoRR*, abs/1511.05644, 2015.

Miyato, Takeru, Kataoka, Toshiki, Koyama, Masanori, and Yoshida, Yuichi. Spectral normalization for generative adversarial networks. 02 2018.

Napoles, Courtney, Gormley, Matthew, and Van Durme, Benjamin. Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, AKBC-WEKEX '12, pp. 95–100, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=2391200.2391218.

Rajeswar, Sai, Subramanian, Sandeep, Dutil, Francis, Pal, Christopher, and Courville, Aaron. Adversarial generation of natural language. *arXiv preprint arXiv:1705.10929*, 2017.

Ranzato, MarcAurelio, Chopra, Sumit, Auli, Michael, and Zaremba, Wojciech. Sequence level training with recurrent neural networks. *arXiv:1511.06732*, 2016.

Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc V. Sequence to sequence learning with neural networks. *arXiv:1409.3215*, 2014.

Villani, Cedric. *Optimal Transport: Old and New.* 2009.

Vincent, Pascal, Larochelle, Hugo, Lajoie, Isabelle, Bengio, Yoshua, and Manzagol, Pierre-Antoine. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408, December 2010. ISSN 1532-4435.

Yu, Lantao, Zhang, Weinan, Wang, Jun, and Yu, Yong. Seqgan: Sequence generative adversarial nets with policy gradient. *CoRR*, abs/1609.05473, 2016.

Zhang, Han, Xu, Tao, Li, Hongsheng, Zhang, Shaoting, Wang, Xiaogang, Huang, Xiaolei, and Metaxas, Dimitris N. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *CoRR*, abs/1710.10916, 2017.

Zhao, J., Kim, Y., Zhang, K., Rush, A., and LeCunn, Y. Adversarially regularized autoencoders. In *eprint arXiv:1706.04223v2*, 2017.

Appendix

A. Progressive Vocabulary Schedules

For direct vocabulary schedules, we used the following cut-offs [1000,2000,5000,8000,11000] i.e. we started with 1000 most frequent words and progressively increased it to 11000 words.

For part of speech based progressive vocabulary, we used the following schedule

1. noun :1000, verb : 500, adjective :200, adverb :50, misc:30, pronoun: 10, others:0
2. noun: 3000, verb: 1000, adjective :400, adverb:100, misc:50, pronoun :20, others:30
3. noun: 5000, verb: 1700, adjective : 800, adverb : 200, misc : 100, pronoun : 44, others:97
4. (full)

B. Qualitative Sentence Sampling(Continued across 4 pages)

Table 2. Sentence generation from noise: To provide a general sense of sentence quality, we sample each model for randomly generated sentences.

ARAE	**ARAE-GP**	**ARAE-GP-NP**
A crowd eats a golf in a guitar .	A clown is asleep by the house bed .	There are women outdoors on the sidewalk during the day .
man is on the waiting	Two brothers are dancing on a couch	The crowd is <oov>by the audience .
The girl is getting wet .	It is raining on a sunny day .	There men no animals .
Two girl playing bread at a small party .	A man is petting a party .	The children are running in a snow , with a large tree .
A cowboy with baseball hat can his band on the stairs .	Man and a woman who are standing up in the man to a bridge .	People are smoking a cigarette stand .
A worker , a purple belt and a red bikini are riding .	An orchestra , woman enjoys the game of others , while on her vacation .	Some children are competing for school .
The worker is heading where a driver	A woman , wearing all a blue t-shirt and a man runs through the field while another man runs to be a long day .	The dogs are running fast .
A man is wearing a black .	A baby is hiding in the sand at a sports match .	The men are surfing to the ground building .
A few man jumped down the streets of music at a tennis party outside .	A man is watching tv .	There were people at a glass counter , eating food .
A man is dressed in ready to two friend get to the hospital a plane he has seen at once .	Two brothers are dancing to the local pool .	Some construction workers are digging through the sand on a playground .

Table 3. Sentence interpolation: To further illustrate the smoothness of the modeled hidden space, we interpolate in the hidden space linearly between two randomly generated sentences. Specifically, we investigate smoothness by sampling two points z_0 and z_1 from $p(z)$ and constructing intermediary points $z\lambda = \lambda z_1 + (1 - \lambda)z_0$, interpolating in the latent space. For each we generate the argmax output. To make models comparable, we also provide aligned samples, in which sentences z_0 and z_1 are fixed, and transformed into each of the models hidden space.

ARAE	**ARAE-GP**	**ARAE-GP-NP**
The man is holding with the bowl .	There are a man dancing on a couch	The men are marching in a club .
The man is holding the bowl .	There are people playing a game	The men are building in a park .
	An infant is playing cards .	The couple are performing at a party .
The boy a father on a hit skateboard is with the violin brother.	An athlete was currently on the swings .	There are people gathered for a girl .
The boy a father on a hit skateboard is with the other infant .	An angry man sat in the classroom .	There are people outside by an animals .
The boy a father on a hit skateboard is with the violin .		
a toddler boy on a stage with dad swings a guitar .	The lead man takes a huge day .	There is a yellow motorcycle parked in traffic .
The boy a father on a hit while another boy is in the living .	The batter is a kite surfer .	Someone fixing a bicycle tire in town .
a toddler boy on a tie with a tennis bat .	The puppy is wearing a bikini .	The car driver crashed in a forest .
	The puppy is playing Frisbee .	The men run quickly in the sand .
The man is a hat that hangs .	The dogs are asleep on three	The men do n't match at play .
The man is a hat that sees .		
	The children are playing tag .	There are an elderly women playing the saxophone .
A dog in a red jacket is pulling a gray cap on a chair pointing.	The boy is playing tag in the sandbox .	There are people having a party at a park .
A dog in a red jacket is pulling a gray cap on a guitar while pointing .	This is is playing in nature .	The couple share a dance at a club .
A dog in a red jacket is pulling a gray cap on a guitar looking pole .	No one is relaxing playing a game .	The men , a woman having a beer with a friend .
	Two dogs are taking a running on a bone .	a couple men and a woman skating while people on the street.

Table 4. Aligned sentence interpolation: To make models more comparable, we also present aligned interpolation. To do this, we encode two given sentences in the code of the respective models. We then interpolate in the hidden space, and decode with the autoencoder. Note that during the encoding and decoding steps, information is lost, so the sentences are not exactly the same between the models, however all codes come from the same input sentences.

ARAE	**ARAE-GP**	**ARAE-GP-NP**
A group of boys who watches a wall performer with an orange artist goes around a cart just had a hard hat while sitting on a motorcycle board .	A man is wearing an orange coat while a woman walks past a building with some sort of water at a mountain .	Two workers digging towards a concrete wall while working .
A group of <oov>men pushing a gym while wearing a purple board to make a large room with a brick bench and some sitting down a road .	A man in an orange shirt is carrying a bag while walking past a yellow building with a lot of cars .	People use ropes while having a picnic under a tent .
A group of <oov>teenager a rail outside an adult and a short lady has been enjoying a night of the street while a red dance works with a band	A man with an American woman and a green shirt walks with a big umbrella outside of the city .	man with an umbrella and a beautiful walk on the beach .
A referee , one older woman is having a conversation and a man with a shopping bag , making the back of a music male with a band sitting on	A man with blond-hair and a dark pink shirt playing with a bunch of the park .	man with three women and two boys behind a bar .
A man , female in a tank top is holding a drum over a shoulder and talking with a young boy and a baby on the back with a laptop	A man with blond-hair and a dark blue shirt sitting with the audience .	man with woman holding hands and talking with flowers inside the kitchen .
A <oov>woman wearing a red skirt and the other are moving the boys line in the girls are playing a car in the park .	Five the kids are smiling at the young male is wearing a red dress in the same outfit .	A man is in the small a truck the the back .
A <oov>woman wearing a red shirt are passing each other the grass to a school in a blue dress and the boys in a black car outfit .	A man in a purple sweatshirt is holding a small <oov>in a red dress , the other team .	A man is drinking a girl in the the race .
A <oov>woman wearing a red shirt looks across the sidewalk as a male officer is wearing a red suit and a <oov>leather jacket in a park .	A man wearing a purple t-shirt is <oov>a small dog in a blue outfit .	A man boy jumps in a white backpack the a bag .
A <oov>woman wearing a bikini top is shopping at a pink stop and a blue purse in a red shirt with a blue umbrella .	A boy wears a white t-shirt and a blue shirt in a blue shirt .	A man jumps a white girl in a wheelchair on a swing .
A <oov>woman wearing a bikini top sits next to a tennis ball wearing a brightly colored shirt a lot with a blue shirt and a blue cap .	A boy wears a white t-shirt suit with a blue shirt .	boy boy watching as a girl in a blue shirt on a playground .
The young guy , the female officer is leaving the car " the " down a street corner the air and a second man who is juggling the ice cream	A woman is the middle of a bar is walking down the street , a young girl is having a tennis tennis racket .	the person walks down a sidewalk in a crowded street near a shop .
The young men , the female work is <oov>the corner and a lady is moving the way to a local street performer in the air .	The woman is the middle of a bar , the other is having a good time and one is holding .	The person walks down the sidewalk of a business near a shop window .
The Two Asian men , the biker is the same direction and is helping the woman in a parking lot because the lady has a shopping cart on it .	The woman is moving the floor and is going to a small child , while the other is working .	The men takes the bus stop and take a break near a store .
The Two women , the older men are leaving the way and the women are putting a pair of the men who has gone on a bench .	The men are moving and a third man is going and the other is working on a bicycle .	The men takes pictures and take a break walk down the sidewalk .
The Two women are sitting behind the <oov>so that the groom are a third and they are wearing a gray jacket and they sit on the steps .	The men are moving and there is a good time and are working on a building .	The men were taking pictures while they wait for friends .
A group of young children are having a bike photo one of the gray biker who does a green jacket and a bicycle tree .	A woman is a little child wearing a stroller and she walks down a sidewalk that looks through a rope .	girl runs off of a concrete .
A Men in a young picture appears to be a farmer trying his father as another woman does the middle of a brown horse looking over .	A man is watching a child who is doing a trick of wood and jumping into a rope .	man runs over an obstacle .
A young couple of people are helping a man who is <oov>a green car and one man crosses a graffiti , the child looks over a bridge .	A man is trying to be a baby and he rides up a ramp in a dirt road .	man is racing <oov>on an escalator .
A man in a female clothing , one is being a man down a passenger because his bike can be a <oov>a bike .	A man is trying to <oov>a bike .	Nobody is trying something to <oov>.
A man in a white t-shirt because it is a well to put a black <oov>, a woman is trying to <oov>a bike .	A man is trying to <oov>a bike .	The person is trying to <oov>something .
A young woman in what appears to a bright bike plays a flute and a female human is playing a white couch	A woman is pushing a small child at a table , a boy is carrying a child and a little child	a man is playing a playing playing a child playing a water
A young woman in what appears to be a golf club a couch is sitting and the grass having a cup	A woman is pushing a baby is walking by a little girl wearing a pink shirt , is a little boy	The man is playing a playing a child playing a guitar .
A young guy , woman in a heavy coat is looking into the air over a local band 's a sports play a room .	A man is pushing a baby is inside wearing a pink t-shirt , is having a long brown dog .	The guy is playing a guitar .
A young guy , working in a glass to be his student is currently riding a skateboard the little girl on a sports stadium .	A man is pushing his daughter is long a little girl , is wearing a long outfit .	The guy is making a basket .
A middle-aged man wearing a green bag is going to the craft station in the foreground and a school wear is working on the craft .	The weather is coming because his mother is practicing a long day in a sports .	The workers is making repairs on street .

Table 5. Noise perturbation experiment: An ideal representation should be relatively robust to small changes through noise in the input (latent space). To test this, we generate samples with additive noise, and see how this affects the model output.

ARAE	**ARAE-GP**	**ARAE-GP-NP**
The man is holding with the bowl .	The dog is relaxing at the game .	There are some trees on the sidewalk in area .
The man is holding the bowl .	The dogs is waiting for the game .	There are some trees on a sidewalk .
	The dogs is waiting at the game .	
The boy a father on a hit skateboard is with the violin brother .	The man is relaxing at the game .	There are dogs rolling around each other in their yard .
The boy a father on a hit skateboard is with the other infant .		There are dogs rolling around each other .
The boy a father on a hit skateboard is with the violin .	A well , three children wait for the band 's time while they play on .	There are dogs who are standing in their race .
a toddler boy on a stage with dad swings a guitar .	The number of women , enjoy the same time while waiting for their game .	There are dogs swimming after each other in the snow .
The boy a father on a hit while another boy is in the living .	A well 3 children enjoy their favorite band play while waiting for game.	
a toddler boy on a tie with a tennis bat .	A well , three children enjoy their favorite band 's play while waiting game .	two men are getting ready inside of building <oov>.
	The number of students were dancing to be playing while the band play.	two men are getting ready inside
The man is a hat that hangs .		two men are getting some water
The man is a hat that sees .	A man sits quietly at his home club .	
	A man sits quietly at his home .	People are gathered inside .
A dog in a red jacket is pulling a gray cap on a chair pointing .		People are standing out .
A dog in a red jacket is pulling a gray cap on a guitar while pointing .		
A dog in a red jacket is pulling a gray cap on a guitar looking pole .		